

## 대체기계가 존재하는 Job Shop 일정계획 환경에서 납기지연을 최소화하는 방법에 관한 연구

### A Study of Job Shop Scheduling for Minimizing Tardiness with Alternative Machines

김기동\*      김재홍\*\*  
Kim, Ki-Dong      Kim, Jae-Hong

#### Abstract

In these days, domestic manufacturers are faced with managerial difficulties such as the increasing competition in their industry and the increasing power of customers. In this situation, they have to satisfy their customers with high quality of their products and meeting due date of their orders. Production of the order within due date is an important factor for improving enterprise competitiveness. The causes of occurrence of tardiness may be wrong product scheduling, unexpected events in field and so on, a way to minimize tardiness is use of alternative machines, overwork, outsourcing and etc..

In this study, we deal with a scheduling problem that can minimize tardiness using alternative machines. This paper provides a mathematical program and a heuristic method for job shop scheduling for minimizing tardiness with alternative machines. And a proposed heuristic method is verified comparing with optimal solution obtained by ILOG CPLEX.

키워드 : 잡샵 일정계획, 대체 기계, 발견적 기법, ILOG CPLEX

Keywords : job shop scheduling, Alternative Machine, Heuristic, ILOG CPLEX

#### 1. 서론

오늘날 국내 제조업체들은 고객의 소비성향과 요구수준의 다양화로 인한 국내외의 경쟁심화와, 시장개방으로 인한 거대 자본 세력의 국내 상륙, 원자재 값의 변동, 첨단 기술의 급진적 발전 등으로 인해 많은 어려움에 직면하고 있다. 날로 다양해지는 소비자들의 요구와 점점 짧아지는 제품의 수명주기는 기업으로 하여금 낮은 가격으로 소비자가 원하는 품질의 제품을 납기에 맞추어 생산할 수 있는 시스템의 도입을 부추기고 있다. 갈수록 치열해지는 기업 간 경쟁에서 도태되지 않기 위해

서는 고객의 요구, 즉 납기일을 만족시키는 것이 중요하다. 특히 주문에 따라 설계, 생산되는 급형이나 중장비, 계절에 따라 수요 변화가 민감한 제품, 월별 또는 분기별로 특정시점에 발행되는 잡지 등을 생산하는 업체의 경우 납기 준수는 기업의 생존과 직결된 중요한 문제이다[6].

실제 생산현장에서 납기일을 준수하는데 장애가 되는 요인은 생산자원의 능력제약을 충분히 고려하지 못하는 잘못된 생산계획, 납기 변경, 긴급주문의 투입, 기계 고장 등이 있다. 이러한 상황에서 납기지연을 최소화 하기위한 방법으로는 대체기계 사용, 초과근무, 외주처리 등이 있다. 만일 납기지연(tardiness)이 발생하면, 납기지연에 따른 경제적 손실과 재공재고의 증가에 따른 제조원가의 상승 및 고객 만족도의 저하로 인한 시장 점유율 감소를 초래한다[1]. 따라서 제조업체는 납기일을 받든

\* 강원대학교 산업공학과 교수, 공학박사

\*\* (주)인버스, ERP 솔루션 팀

시 지켜야만 하는 상황이다.

본 연구는 공정에 대한 대체 기계에 있어서 제품이 납기일을 만족시킬 수 있는 실제의 기업 환경을 대상으로 하고 있다. 문제의 상황은 일정계획 문제 중 가장 일반적이라고 할 수 있는 job shop 형태이며, 각 공정을 가공할 수 있는 기계가 다수 개 존재하는 상태이다. Job shop 일정계획 문제는 풀기 어려운 조합최적화 문제(combination optimization problem)중에서도 매우 어려운 NP-hard 문제로 알려져 있다[7]. 이 문제를 풀기가 매우 어려움에도 불구하고, 그 중요성 때문에 최적해가 아니더라도 비교적 좋은 해를 구하기 위한 많은 연구가 있었고, 현재에도 진행되고 있다 [2]. 기존의 job shop 일정계획에 관한 연구들의 목적은 최종완료시간(makespan)의 최소화가 대부분이었다[5, 6]. 그러나 최종완료시간이 아무리 짧아도 납기일을 지키지 못한다면 아무런 소용이 없을 것이다. 납기 지연이 발생하게 되면 그에 따른 경제적 손실과 고객 만족도의 저하로 인한 시장 점유율 감소 등으로 인해 크나큰 손실이 발생한다. 그래서 본 연구에서는 고객과의 약속인 납기일을 준수하여 납기 지연을 최소로 하는 것을 목적으로 정하였다.

본 연구는 대체기계가 있는 job shop 환경에서, 납기 지연 최소화를 위한 일정계획을 수립할 수 있는 발견적 기법을 제시한다. 또한 제시된 발견적 기법의 효율성을 입증하기 위해 ILOG CPLEX를 이용하여 구한 최적해와 비교 분석을 실시한다.

2장에서는 수리모형 제시와 ILOG CPLEX를 사용한 해법을 제시하였다. 3장에서 문제해결을 위해 유전 알고리즘을 이용한 발견적 기법을 제시하였고, 4장에서는 개발된 발견적 기법을 이용하여 구한 해를 ILOG CPLEX를 이용하여 구한 최적해와 비교 분석을 실시하여 제시된 발견적 기법의 효율성을 입증하였다. 5장에서 결론 및 추후 연구 과제를 제시하였다.

## 2. 대체기계가 존재하는 job shop 일정계획 문제

전통적인 job shop 일정계획 문제는 m대의 기계에서 서로 다른 가공순서와 가공시간을 가지는 n개의 작업을 처리하는 순서를 결정하는 것이다. Job shop 일정계획 문제에는 크게 기술계약(technical constraint)과 자원계약(resource constraint)의 두 가지의 제약조건이 있다. 기술계약은 일련의 작업순서를 가지고 있는 각 작업의 공정이 선행공정이 완료되기 전까지 시작할 수 없다는 제약이고, 자원계약은 기계가 어떤 시점에서 두 가지 이상의 공정을 동시에 수행할 수 없고 공정수행 도중에 다른 공정을 시작할 수 없다는 제

약이다. Job shop 일정계획 문제는 이러한 제약을 지키며 각 기계에 대한 작업물의 시작 및 완료 시각을 계획하는 것이다[7]. 작업의 가공순서와 이들을 가공하는 기계가 서로 다른 job shop에서는 작업 별 가공시간이 일정하지 않으며, 납기도 서로 다를 수 있다. 또한 job shop에서는 기술적으로 실행 가능한 스케줄의 대안이 대단히 많고, 품목의 종류와 가공에 필요한 생산설비가 많은 경우에는 간단하게 최적 스케줄을 결정하는 것이 매우 어렵다.

전통적인 job shop 일정계획 문제에서 각 작업이 가공 되어야 하는 공정들의 순서가 정해져 있고, 작업의 각 공정이 처리되어야 하는 기계가 정해져 있다는 것을 가정하였는데, 이는 제품이나 작업 부하량 또는 기계 고장과 같은 많은 요인들에 의한 변화에 빠르게 적응하기가 어렵다. 따라서 본 연구에서는 작업의 각 공정을 수행할 수 있는 대체기계가 있다는 것을 가정하여 변화에 대처하려 한다. 현실적으로 자본력이 약한 중소기업은 과거의 전통적인 job shop 형태의 제조시스템을 유지하면서 많은 요인들의 변화에 빠르게 적응하여야 하기 때문에, 본 연구는 대체기계가 있는 job shop 일정계획 문제를 다루려 한다. 대체기계는 단일 목적의 기계가 여러 대 있는 경우와 여러 가지 일을 할 수 있는 다중 목적 기계인 경우가 있는데[3], 본 연구에서는 전자인 상황을 가정한다.

### 2.1 수리 모형

본 절에서는 대체기계가 존재하는 job shop 일정계획 문제의 가정과 수리모형에 대해 설명한다.

#### 가정

- 각 공정은 하나 이상의 대체기계를 가지고 있다.
- 각 공정은 그것을 수행하는 동안 중단되어 지지 않는다.(비선점조건)
- 각 기계는 한 시점에 하나의 공정만을 수행할 수 있다.
- 각 작업의 준비시간 및 이동시간은 가공시간에 포함된다.
- 목적은 가장 작은 납기 지연을 가지는 일정을 찾는 것이다. 여기서 납기 지연은 각 작업이 완료되는 시각에서 각 작업의 납기일을 빼어 양수가 되는 수이다.(음수일 경우 납기 지연은 0)
- 조기완료에 대한 penalty는 없다.

기호 정의

$i$  : 작업 번호(1~N)

$m$  : 기계 번호(1~M)

$d_i$  :  $i$  번 작업의 납기일

$G_i$  :  $i$  번 작업의 공정 수

$g$  : 공정 번호(1~ $G_i$ )

( $i,g$ ) :  $i$  번 작업의  $g$ 번째 공정

$K_{(i,g)}$  : ( $i,g$ )의 대체기계 집합

$P_{igk}$  : ( $i,g$ )를 ( $i,g$ )의 대체기계 중  $k$  기계에서 가  
공할 때 가공시간

$|K_{(i,g)}|$  : ( $i,g$ )의 대체기계 수

$H$  : big number

결정변수

$t_i$  :  $i$  작업의 납기지연

$x_{ig}$  : ( $i,g$ )의 가공완료시간

$y_{igkjh}$  : 1 - 같은 기계  $k$ 에서 ( $i,g$ )가 ( $j,h$ )를 선행  
할 때  
0 - otherwise

$v_{igk}$  : 1 - ( $i,g$ )를  $k(i,g)$ 가 가공할 때  
0 - otherwise

수리모형

$$\text{Minimize } \sum_{i=1}^N t_i \quad (1)$$

subject to

$$x_{iG_i} - d_i \leq t_i, \text{ for } i=1, \dots, N \quad (2)$$

$$x_{i(g+1)} - x_{ig} \geq \sum_{k \in K_{(i,g+1)}} P_{i(g+1)k} v_{i(g+1)k},$$

$$\text{for } i=1, \dots, N, g=1, \dots, G_i-1 \quad (3)$$

$$x_{ig} - x_{jh} + H(2 - v_{igk} - v_{jhk} + y_{igkjhk})$$

$$\geq \sum_{k \in K_{(i,g)}} P_{igk} v_{igk}$$

for  $i, j=1, \dots, N(i \neq j), g, h=1, \dots, G_i \quad (4)$

$$x_{jh} - x_{ig} + H(3 - v_{igk} - v_{jhk} - y_{igkjhk})$$

$$\geq \sum_{k \in K_{(j,h)}} P_{jhk} v_{jhk}$$

for  $i, j=1, \dots, N(i \neq j), g, h=1, \dots, G_i \quad (5)$

$$x_{i1} \geq \sum_{k \in K_{(i,1)}} P_{i1k} v_{i1k}, \text{ for } i=1, \dots, N \quad (6)$$

$$\sum_{k \in K_{(i,g)}} v_{igk} = 1,$$

$$\text{for } i=1, \dots, N, g=1, \dots, G_i \quad (7)$$

식(1)은 목적함수으로써 각 작업의 납기지연의 총합 최소를 의미한다.  $t_i$ 는 비음계약으로 인해 0보다 크거나 같은 값을 갖는다. 식(2)는 납기지연에 대한 정의를 나타낸 제약식이다. 식(3)은 작업 내에서의 공정 간의 선후관계에 대한 제약조건을 표현한 식이다. 식(4), 식(5)는 한 기계 내에서의 작업 간의 선후관계에 대한 제약조건을 표현한 식으로써 식(4)는 ( $i,g$ )공정이 선행할 경우이고, 식(5)는 ( $j,h$ )공정이 선행할 경우를 나타낸 식이다. 한 기계에서 동시에 두 개의 공정을 수행할 수 없다는 의미이다. 식(6)는 각 작업의 첫 번째 공정의 완료시각은 그 가공시간보다 크거나 같다는 제약을 표현한 식이다. 식(7)은 하나의 공정은 그 공정의 대체기계 중에서 단 하나의 기계에서만 가공을 해야 한다는 제약을 표현한 식이다. 마지막으로 모든 변수는 음수가 아니다.

2.2 ILOG CPLEX를 사용한 해법

제약만족 기법(Constraint Satisfaction Technique : CST) 개념은 주어진 제약조건을 만족하는 해를 찾아나가는 방법을 가리킨다. 즉, 제약만족 기법은 문제를 구성하는 제약조건과 제약변수를 가지고 문제를 모델링한 후 여러 가지 방법으로 제약조건을 만족하는 제약변수의 값을 구하고자 하는 방법이다. 본 연구에서는 대체기계가 있는 job shop 일정계획 문제가 가지고 있는 변수와 제약을 이용하여 ILOG CPLEX 문제로 표현하였다. 일정계획 문제에서 결정할 변수(작업 종료시간, 작업 기계 등)와 이 변수의 값을 제한하는 제약(작업 순서, 사용가능기계, 성과측정치 등)을 설정하고, 이 제약을 만족하는 변수 값을 찾을 것이다.

본 연구에서 사용하는 ILOG 제품군은 CSP(Constraint Satisfaction Problem)를 기반으로 하는 프로그래밍 라이브러리로 통신, 제조, 운송 등의 분야에서 발생하는 자원제약 문제를 해결하기 위한 제품으로써 문제를 해결하는 부분인 Optimization 제품군(ILOG Solver, Scheduler, CPLEX)과 사용자와 인터페이스를 위한 Visualization 제품군(ILOG Views)으로 이루어져 있다. 본 연구에서는 Optimization 제품군 중 ILOG CPLEX 7.1 version을 사용하였다.

(1) 결정변수

본 연구의 결정변수들을 다음과 같이 표현하였다. nbJob, nbProcess, nbMachine은 작업의 수, 공정의 수, 기계의 수이다. 아래의 표현들은 각각의 변수들의 도메인 값을 지정해 준 것이다.

i)  $t_i$  :  $i$  작업의 납기지연  
 $t_i$ 를 숫자변수(NumVar)배열로 지정하여 0부터 무한대까지의 값을 갖도록 도메인 값을 지정하였다.  
 IloNumVarArray t(env, nbJob, 0, IloInfinity);

ii)  $x_{ig}$  :  $(i,g)$ 의 가공완료시간  
 $x_{ig}$  역시 숫자변수배열로 지정하고 작업(i)에 따라 공정 수만큼의 배열을 생성하였다. 이 역시 0부터 무한대까지의 값을 갖고 있다.  
 $X[i] =$  IloNumVarArray(env, nbProcess, 0, IloInfinity);

iii)  $y_{igkjhk}$  : 1 - 같은 기계 k에서  $(i,g)$ 가  $(j,h)$ 를 선행할 때,  
 0 - otherwise  
 $y_{igkjhk}$ 는 숫자변수배열로 지정하고, 작업(i, j), 공정(g, h)에 따라 기계 수 만큼의 배열을 생성하였다. 배열 안에는 0과 1이 값이 들어가는 0/1변수이고 자료형은 Int형이다.  
 $Y[i][g][j][h] =$  IloNumVarArray(env, nbMachine, 0, 1, ILOINT);

iv)  $v_{igk}$  : 1 -  $(i,g)$ 를 k( $i,g$ )가 가공할 때,  
 0 - otherwise  
 $v_{igk}$  역시 0/1변수로써 작업(i)과 공정(g)에 따라 기계 수 만큼의 배열을 생성하였다.  
 $V[i][g] =$  IloNumVarArray(env, nbMachine, 0, 1, ILOINT);

(2) 제약의 표현

수리모형에 나온 각각의 제약들을 본 연구에서는 다음과 같은 방식으로 표현하였다. model.add를 이용하여 각각의 제약을 추가하였다.

i) 1번식  
 본 연구의 목적함수로써 납기지연의 합(IloSum(t))을 IloMinimize를 사용하여 최소화 시켰다.  
 model.add(IloMinimize(env, IloSum(t)));

ii) 2번 제약식  
 각 작업의 마지막 공정인  $G_i$ 를 for문이 0부터 돌기 때문에 총 공정의 수(nbProcess)에서 1을 빼어 다음과 같이 표현하였다.  
 model.add( $X[i][nbProcess-1] - df[i] \leq t[i]$ );

iii) 3번 제약식  
 우선 3번 제약식의 우변의  $P_{i(g-1)k}V_{i(g-1)k}$ 를 sum\_p[k]에 저장하는 제약과 sum\_p[k]의 합을 다시 제약으로 추가하여 다음과 같이 표현하였다. p\_time에는 각 공정에 할당된 대체기계의 가공시간이 들어있다.

model.add(sum\_p[k] == p\_time[i][g+1][k] \*  $V[i][g+1][k]$ );  
 model.add( $X[i][g+1] - X[i][g] >=$  IloSum(sum\_p));

iv) 4번 제약식  
 4번 제약식 우변의  $P_{igk}V_{igk}$ 를 sum\_ig에 누적시킨 후 각각 제약에 추가하여 다음과 같이 표현하였다.  
 $sum\_ig += p\_time[i][g][k]*V[i][g][k]$ ;  
 model.add( $X[i][g] - X[j][h] + 10000*(2 - V[i][g][k] - V[j][h][k] + Y[i][g][j][h][k]) >=$  sum\_ig);

v) 5번 제약식  
 5번 제약식 우변의  $P_{jkh}V_{jkh}$ 를 sum\_jh에 누적시킨 후 각각 제약에 추가하여 다음과 같이 표현하였다.  
 $sum\_jh += p\_time[j][h][k]*V[j][h][k]$ ;  
 model.add( $X[j][h] - X[i][g] + 10000*(3 - V[i][g][k] - V[j][h][k] - Y[i][g][j][h][k]) >=$  sum\_jh);

vi) 6번 제약식  
 좌변의  $P_{ik}V_{ik}$ 를 sum\_V[k]에 저장한 후 sum\_V[k]의 합이  $x_{i0}$ 보다 작거나 같다는 제약을 주었다.  
 model.add(sum\_V[k] == p\_time[i][0][k] \*  $V[i][0][k]$ );  
 model.add( $X[i][0] >=$  IloSum(sum\_V) );

vii) 7번 제약식  
 $v_{igk}$ 의 합을 sum\_V에 저장한 후 sum\_V가 1과 같다는 제약식을 표현한 것이 다음과 같다.  
 $sum\_V += V[i][g][k]$ ;  
 model.add( sum\_V == 1);

3. 문제 해결을 위한 발견적 기법

Job shop 일정계획 문제에서 실제 문제를 해결하기 위한 시간과 자원들은 매우 부족하나, 여전히 의사결정자들은 좋은 해를 요구하기 때문에 휴리스틱 방법들이 유력한 대안이 되고 있다[4]. 1980년대 이후 타부 서치, 시뮬레이티드 어닐링, 유전 알고리즘 등의 반복적 알고리즘에 관한 연구가 활발히 이루어지고 있다. 그들 각자는 좋은 국소 최적해를 찾기 위해 문제의 해 영역들을 탐색해 가는 전략들을 정의하고 있다. 본 연구에서는 적자생존이라는 진화개념을 사용하는 유전 알고리즘을 job shop 일정계획 문제의 해법으로 사용하고자 한다.

### 3.1 해의 표현

유전 알고리즘으로 job shop 일정계획 문제를 해결하기 위해서는 먼저 문제의 해를 염색체로 표현해야 한다. 기존의 유전 알고리즘에서의 해의 표현은 연산자를 사용하여 가능해 공간을 효율적으로 탐색할 수 있도록 이진수로 표현하고 있으나, job shop 일정계획 같은 문제는 이진수로 표현하기 어렵고 연산 시에 가능해를 유지하기가 매우 어렵다. 따라서 job shop 일정계획 문제에서는 해의 실행 가능성을 유지하는 것이 매우 중요하고, 이는 주로 적절한 해 표현 방법의 설계를 통해서 해결할 수 있다. 기존의 유전 알고리즘에서처럼 이진 표현방법을 적용하지 않고, 본 연구에서는 다음과 같은 방법으로 해 표현 방법을 설계하였다.

각 개체는 문제에 대한 해를 나타낸다. 개체는 하나 이상의 염색체로 구성되어질 수 있고, 스트링(string)에 의해 표현된다. 해의 표현 방식은 유전 알고리즘의 성공적인 수행을 위한 매우 중요한 단계이자, 첫 번째 단계이다. 우선 염색체의 번호는 인덱스(Index) 번호를 의미하는데, 인덱스 밑에 작업, 공정, 기계, 가공시간의 정보가 들어간다. 따라서 다음의 표 1과 같이 인덱스 번호 하나만으로 모든 정보를 나타낼 수가 있는 것이다.

표 1 인덱스 설명(2작업×3공정×4기계의 예)

인덱스	1	2	3	4	5	6	7	8	9
작업	1	1	1	1	1	1	1	1	1
공정	1	1	1	2	2	2	3	3	3
기계	1	2	4	2	3	4	1	3	4
가공시간	4	5	6	1	6	8	2	3	9

인덱스	10	11	12	13	14	15	16	17	18
작업	2	2	2	2	2	2	2	2	2
공정	1	1	1	2	2	2	3	3	3
기계	2	3	4	1	2	4	1	2	3
가공시간	1	3	5	2	2	7	6	2	5

각 개체는 두 개의 염색체(염색체 A, 염색체 B)로 구성이 되는데, A 염색체는 각 공정에 할당된 인덱스 번호를 나타내고 B 염색체는 각 기계에 할당된 인덱스 번호의 가공순서를 나타낸다. 예를 들어 A 염색체와 B 염색체를 나타내면 다음 표 2, 표 3과 같다.

표 2 A 염색체의 예

(작업, 공정)	(1,1)	(1,2)	(1,3)	(2,1)	(2,2)	(2,3)
인덱스	1	5	9	11	13	17

표 3 B 염색체의 예

기계	인덱스	
M <sub>1</sub>	1	13
M <sub>2</sub>	17	
M <sub>3</sub>	5	11
M <sub>4</sub>	9	

### 3.2 초기해 생성

국소 탐색법에서 초기해의 값은 해에 많은 영향을 미친다. 본 연구에서는 우수한 개체로 구성된 초기 모집단을 생성시키기 위하여 A 염색체와 B 염색체의 초기해를 다음과 같은 방법으로 생성하였다.

#### (1) A 염색체 초기해 생성

각 공정의 대체기계 중에서 가공시간이 가장 짧은 기계를 확률적으로 선택이 잘 되도록 한다. 예를 들어 설명하면 다음 표 4에서 1번 작업의 1번 공정의 대체기계는 1, 2, 3번 기계이다. 이들의 가공시간의 역수 1/4, 1/5, 1/6을 모두 더하면 0.61이 된다. 다시 가공시간의 역수를 0.61로 나누면 0.41, 0.33, 0.26이 되는데 이는 각 기계가 선택될 확률이다. 이들의 누적확률을 구하면 0.41, 0.74, 1이 된다. 그 후에 [0, 1] 구간 내의 값을 갖는 난수를 생성하는데, 임의의 난수 r1의 값이 0.3이라고 하면 0 < 0.3(r1) < 0.41이 되므로 1번 작업의 첫 번째 공정의 기계는 첫 번째 기계인 1번 기계가 선택되어 인덱스 번호 1이 제일 먼저 할당된다. 이러한 초기해 생성방법으로 가공시간이 짧은 기계의 선택될 가능성을 확률적으로 높일 수 있었다. 이 과정이 끝나면 A 염색체의 해 집단크기만큼 초기해 집단이 생성되게 된다.

표 4 A 염색체 생성의 예제

인덱스	1	2	3
작업	1	1	1
공정	1	1	1
기계	1	2	3
가공시간	4	5	6

#### (2) B 염색체 초기해 생성

생성된 각각의 A 염색체에는 그에 따르는 매우 많은 조합의 B 염색체가 생성될 수 있다. A 염색체 하나로 각 공정에 할당된 기계의 정보를 얻을 수 있지만, 각 기계에 할당된 인덱스 번호의 순서에 대한 정보는 얻을 수 없다. 그래서 본 연구에서는 A 염색체 하나당 가장 좋은 B 염색체를 찾기 위하여 내부 유전 알고리즘을 실시하였는데, 그러기 위해서는 내부 유전 알고리즘에서 필요한 B 염

색체의 초기해의 생성이 필요하다. B 염색체의 초기해는 A 염색체의 정보에 따라 기계에 각 인덱스 번호를 할당한 후, 각 기계 상에서 랜덤하게 인덱스 번호의 가공 순서를 정한다. 그러면 B 염색체의 선택 및 유전 연산(교배, 돌연변이)을 위한 초기해가 생성된다.

### 3.3 선택

목적함수에 따라 평가 값이 계산되면 다음 세대를 생성하기 위해 집단 내에서 우수한 개체를 선택한다. 이 때, 우수한 개체 일수록 높은 선택확률을 갖도록 하는 것이 선택 방법 결정의 핵심이라고 할 수 있다. 이러한 관점에서 평가 값의 확률적 분포를 반영하여 새로운 집단을 생성하는 가장 일반적인 방법은 룰렛 휠 표본 추출(Roulette wheel sampling)방법이다. 룰렛 휠 방법은 다음과 같은 절차로 진행된다.

- 목적함수에 따른 평가 값의 총합을 계산한다.
- 각 개체에 대해 선택확률을 계산한다.
- 각 개체에 대해 누적확률을 계산한다.
- [0,1] 구간 내의 값을 갖는 난수를 생성한다.
- 생성된 난수가 포함되는 누적확률을 갖는 개체를 선택한다.

본 연구의 목적함수가 납기지연의 최소화이므로 각각의 해에 대한 선택확률을 구할 때는 평가 값의 역수를 사용한다(단, 평가 값이 0일 경우 optimal 이므로 임의의 값을 적용). 이 과정을 거치면 기존에 생성된 A, B 염색체 초기해 집단이 우수한 평가 값을 지닌 집단으로 재생성 된다.

### 3.4 교배

교배는 자연계에서 생식에 의하여 새로운 개체가 생성되는 개념을 도입한 것으로, 교배의 역할은 현재의 좋은 개체에 포함되어진 정보를 교환함으로써 더 나은 해를 생성하는 것이다. 본 연구에서는 A 염색체와 B 염색체의 교배를 각각 다르게 적용한다.

#### (1) A 염색체의 교배

A 염색체의 교배 연산은 두 개의 A 염색체에 대해 수행되는데, 랜덤하게 교배 구간이 결정된다. 선택된 두 구간의 인덱스 번호들을 바꿔주면 새로운 자식 염색체가 생성된다. 그 과정은 다음 그림 1과 같다. A 염색체의 교배가 끝난 후에는 그에 따른 B 염색체도 수정을 해준다.

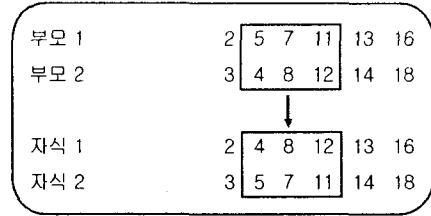
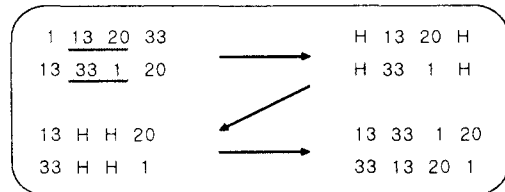
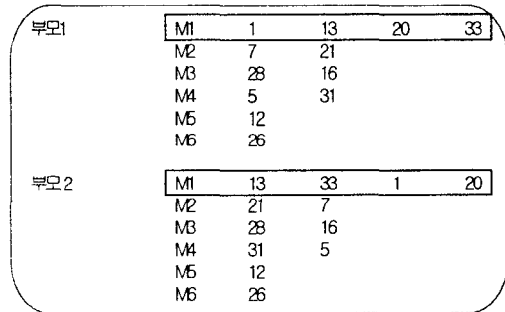


그림 1 A 염색체의 교배

#### (2) B 염색체의 교배

B 염색체의 교배 연산 역시 두 개의 B 염색체에 대해 수행되는데, 랜덤하게 교배 대상, 교배 기계와 교배 구간이 결정된다. 할당된 공정이 3개 이상인 기계들 중에서 공정 수가 많은 기계가 돌연변이 대상으로 잘 설정되도록 하였다. B 염색체 교배는 A 염색체처럼 하면 같은 인덱스 번호가 같은 기계에 동시에 할당되는 일이 발생할 수 있기 때문에 LOX(Linear Order Crossover)방법으로 교배를 실시한다. 그 과정은 다음 그림 2와 같다. 단, 교배 결과 실현 불가능한 자식 해들이 나오면 교배 이전으로 다시 되돌리는 연산을 실시한다. 그림 2를 보면 기계 1번이 교배 대상 기계로 선택되었고, 기계 1번의 인덱스 번호들이 교배가 되어 새로운 자식 개체들이 형성되는 것을 볼 수가 있다.



자식1	M1	13	33	1	20
	M2	7	21		
	M3	28	16		
	M4	5	31		
	M5	12			
	M6	26			
자식2	M1	33	13	20	1
	M2	21	7		
	M3	28	16		
	M4	31	5		
	M5	12			
	M6	26			

그림 2 B 염색체의 교배

### 3.5 돌연변이

돌연변이는 우선 한 세대를 형성하는 개체의 다양성을 유지하게 하여 국부 최적해에 빠질 가능성을 더욱 낮추는 역할을 함과 동시에, 돌발적으로 우수한 개체가 생성될 가능성을 제공한다.

#### (1) 염색체 A의 돌연변이

A 염색체에 대한 돌연변이 연산은 할당된 인덱스 번호를 재지정하는 방법으로 한다. 우선 돌연변이 위치를 랜덤하게 선택한 후, 현재의 인덱스 번호를 대체 가능한 같은 작업의 같은 공정의 임의의 다른 인덱스 번호로 할당하는 것이다. 다음 그림 3과 같이 랜덤하게 인덱스 번호 11, 16을 선택하고, 가능한 인덱스 번호인 10, 18로 바꿔 주는 것이다. A 돌연변이가 끝난 후에는 그에 따른 B 염색체도 수정해준다.

돌연변이 전	2	5	7	11	13	16
				↓		↓
돌연변이 후	2	5	7	10	13	18

그림 3 A 염색체의 돌연변이

#### (2) B 염색체의 돌연변이

B 염색체의 돌연변이 연산은 랜덤하게 돌연변이 대상 염색체, 기계, 돌연변이 구간이 결정된다. 할당된 공정이 3개 이상인 기계를 중에서 공정수가 많은 기계가 돌연변이 대상으로 잘 설정되도록 하였다. B 염색체의 돌연변이는 돌연변이 구간으로 설정된 구간의 인덱스 번호의 순서를 역으로 바꾸어 주는 것으로, 돌연변이 대상 기계가 1번으로 선택되었을 때, 다음 그림 4와 같다. B 염색체의 돌연변이 후 교착상태에 빠진 불가능해가 나오면 돌연변이 이전의 원래 염색체로 환원해준다.

돌연변이 전	M1	1	13	20	33
	M2	7	21		
	M3	28	16		
	M4	5	31		
	M5	12			
	M6	26			
돌연변이 후	M1	1	33	20	13
	M2	7	21		
	M3	28	16		
	M4	5	31		
	M5	12			
	M6	26			

그림 4 B 염색체의 돌연변이

### 3.6 교착상태(dead lock) 처리

유전 알고리즘을 수행하다보면 초기해 생성이나 연산부분에서 교착상태에 빠지는 해가 나올 수가 있다. 교착상태란, 모든 공정이 선행공정을 가지고 있어 어느 공정도 스케줄링 할 수가 없는 상황을 말한다. 따라서 교착상태에 빠진 해는 평가 값도 구할 수가 없다. 다음 그림 5의 상황이 작업 4개, 공정 3개, 기계 6대의 상황에서 교착상태에 빠진 해의 예이다. 4번 작업의 첫 번째 공정을 1번 기계에서 스케줄하려고 하면 그 공정의 작업 상 선행공정인 4번 작업의 2번째 공정을 미리 스케줄 하여야만 한다. 그러나 4번 작업의 2번째 공정은 2번 기계에서 1번 작업의 2번째 공정이 끝나야만 스케줄을 할 수가 있다. 1번 작업의 2번째 공정은 1번 작업의 1번째 공정을 스케줄한 후에야 할 수 있는데, 1번 작업의 1번째 공정은 4번 작업의 3번째 공정이 끝난 후에야 1번 기계에서 스케줄이 가능하다. 이처럼 교착상태에 빠진 4개의 공정들은 어느 하나도 스케줄을 할 수가 없다.

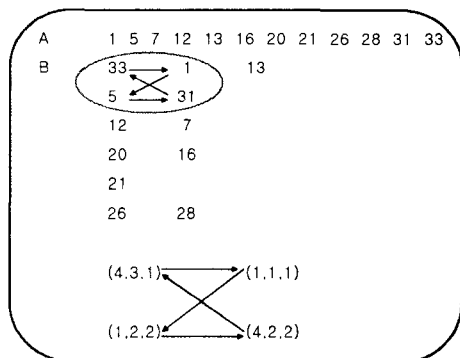


그림 5 교착상태의 예

위의 예와 같은 B 염색체의 교착상태는 알고리즘 수행 과정에서 제외시켜야 한다. 스케줄이 가능한 염색체들을 스케줄해가다가 선행 공정이 있어서 더 이상 스케줄이 불가능하여 모든 염색체를 스케줄하지 못하는 상황, 즉 교착상태에 빠지면 그 해는 제외시킨다. 그리고 교배나 돌연변이 이전의 해로 환원시킨다. 본 연구에서는 이렇게 교착상태에 빠진 해를 검출하여 추후 알고리즘 수행과정에서 제외하였다.

### 3.7 유전 알고리즘 구조

다음 그림 6은 대체기계가 있는 job shop 일정 계획 문제를 풀기 위한 유전 알고리즘의 전체적인 흐름을 보여주고 있다. 우선 A 염색체의 초기해를 생성하고 각각의 A 염색체 하나당 가장 좋은 B 염색체를 찾기 위하여 1번 loop를 수행한다. A 염색체 하나에 따라 B 염색체의 초기 모집단을 B 염색체의 해 집단크기만큼 생성한 후, 평가 값에 따라 우수한 평가 값을 갖는 해 집단을 재생성한다. B염색체의 교배 후 불가능한 해가 나오면 교

배 전의 부모해로 환원 시키고 돌연변이를 실시한다. B 염색체 돌연변이도 실시 후에 불가능한 해가 나오면 돌연변이 전의 염색체로 환원시킨다. 1번 루프는 기본적으로 세대수만큼 돌게 되는데, 도중에 optimal(납기지연이 0)이 나오거나, 잇따르는 지정횟수에서 해의 개선이 보이지 않으면 탈출한다. 이 과정을 거치면 각 A 염색체에는 B 염색체가 하나씩 생기게 된다. B 염색체의 평가 값이 해당하는 A 염색체의 평가 값이 되는데 이를 바탕으로 A 염색체의 선택과 재생성에 들어가게 된다. 선택은 평가 값을 이용하여 룰렛 휠 방법으로 비교적 좋은 해를 가진 집단을 만든다. A 염색체 교배 후에는 그에 딸린 B 염색체를 수정하고 돌연변이에 들어간다. A 염색체 돌연변이 역시 완료 후에 그에 딸린 B 염색체를 수정한다. A 염색체에 관련된 2번 루프도 지정 세대수만큼 돌다가 optimal이 발견되거나, 잇따른 지정횟수에서 해의 개선이 보이지 않으면 탈출한다. 본 연구에서는 해의 개선이 이루어지지 않고 도는 최대 수를 5회로 지정하고 실험을 하였다.

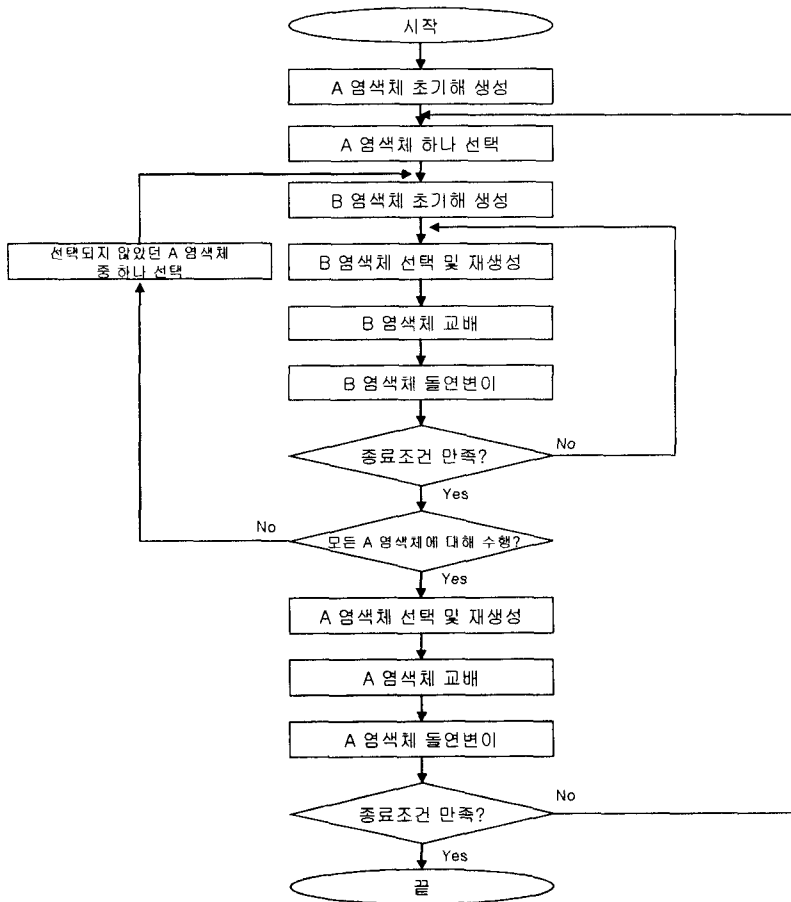


그림 6 유전 알고리즘 흐름도



#### 4. 실험 결과 및 분석

본 연구에서는 제시한 유전 알고리즘의 효율성을 입증하기 위하여 실험을 실시하였다. 유전 알고리즘의 효과 분석을 위해 동일한 목적함수 하에서 ILOG CPLEX를 이용하여 문제를 풀이한 결과와 유전 알고리즘을 이용하여 문제를 풀이한 결과를 비교 분석하였다. 실험에 사용한 입력 데이터는 작업 4개, 공정 3개, 기계 6대가 있는 경우(이하 4×3×6)와 작업 8개, 공정 3개, 기계 8대(이하 8×3×8)가 있는 경우로 작은 규모의 문제와 비교적 큰 규모의 문제로 나누어서 실험을 실시하였다. 각각의 문제별로 50개의 데이터를 랜덤하게 생성하였고, 생성된 데이터를 바탕으로 실험을 실시하였다. 4×3×6 문제를 풀기 위한 유전 알고리즘 실험에 사용한 각종 파라미터 값들은 다음 표 5에 나와 있다. 실험에 사용한 컴퓨터는 팬텀 IV 3.4G, 메모리 1G의 기종이다. 제시한 유전 알고리즘과 ILOG CPLEX를 이용하여 구한 해를 비교하기 위하여 4×3×6 문제의 실험을 실시한 결과는 다음 표 6의 비교 실험결과 1과 같다.

표 5 유전 알고리즘 파라미터

A 염색체의 해집단 크기	50
B 염색체의 해집단 크기	20
A 염색체의 교배율	0.5
B 염색체의 교배율	0.5
A 염색체의 돌연변이율	0.2
B 염색체의 돌연변이율	0.2
A 염색체의 총 세대수	100
B 염색체의 총 세대수	10
A 염색체의 해 개선이 이루어지지 않고 도는 최대 수	5
B 염색체의 해 개선이 이루어지지 않고 도는 최대 수	5

표 6 비교 실험 결과 1

	ILOG CPLEX		유전 알고리즘		최적해와의 차이
	계산 시간(초)	납기 지연	계산 시간(초)	납기 지연	
1	0	0	6	0	0
2	0	0	5	0	0
3	0	0	9	0	0
4	0.02	0	26	0	0
5	0	0	23	3	3
6	0	0	69	1	1

7	0	0	48	5	5
8	0.02	0	5	0	0
9	0	0	78	0	0
10	0	0	13	0	0
11	0	6	29	27	21
12	0	0	18	0	0
13	0	0	11	0	0
14	0	0	6	0	0
15	0	0	10	0	0
16	0	0	10	2	2
17	0	0	20	0	0
18	0	0	28	1	1
19	0	2	57	5	3
20	0	0	3	18	18
21	0	0	28	0	0
22	0	0	26	0	0
23	0	0	8	8	8
24	0	0	7	5	5
25	0	0	14	0	0
26	0	0	33	2	2
27	0	0	4	5	5
28	0	1	11	36	35
29	0	0	12	14	14
30	0	0	23	7	7
31	0	0	4	2	2
32	0	0	8	19	19
33	0	0	5	0	0
34	0.02	0	7	0	0
35	0.02	2	12	6	4
36	0	0	9	0	0
37	0	0	12	1	1
38	0	0	1	2	2
39	0	0	8	1	1
40	0	0	10	7	7
41	0.02	0	11	1	1
42	0	0	5	1	1
43	0	0	5	1	1
44	0	0	5	0	0
45	0.01	0	6	0	0
46	0	1	11	9	8
47	0	0	15	0	0
48	0	0	2	3	3
49	0	0	9	1	1
50	0	0	7	0	0
평균	0.002	0.24	15.84	3.86	3.62

표 6의 결과를 보면 ILOG CPLEX로 문제를 풀이한 평균 계산 시간이 유전 알고리즘으로 풀었을 경우보다 좀 더 적게 걸린 것을 알 수 있다. 유전 알고리즘이 ILOG CPLEX로 구한 최적해를 찾을 확률은 49%였고, 최적해와 평균 3.62만큼 해의 차이를 보였다.

좀 더 규모가 큰 문제의 경우를 살펴보기 위해

다음 8×3×8 문제의 실험을 유전 알고리즘과 ILOG CPLEX 별로 각각 실시하였다. 8×3×8 문제의 실험 결과는 다음 표 7과 같다. 아래의 결과를 보면 비교적 규모가 큰 문제(8×3×8)의 경우, ILOG CPLEX는 최적해를 산출하기는 하였으나 시간이 오래 걸렸으며, 유전 알고리즘이 ILOG CPLEX로 구한 최적해를 찾지는 못하였지만, 비교적 빠르게 해를 산출해 내었다는 것을 알 수 있다. 이와 같이 규모가 큰 문제의 경우에는 ILOG CPLEX로 문제를 푸는 것이 유전 알고리즘으로 푸는 것에 비해 시간이 매우 오래 걸리는 것을 알 수 있었다. 따라서 문제 규모가 더 크고, 실시간으로 생산 스케줄을 짜야 하는 실제 생산 현장에서는 본 연구에서 제시한 유전 알고리즘을 이용한 발전적 기법을 사용하는 것이 현장에서 발생하는 변화에 대처하기에 더 유리하다고 할 수 있다.

표 7 비교 실험 결과 2

	ILOG CPLEX		유전 알고리즘		최적해와의 차이
	계산 시간(초)	납기 지연	계산 시간(초)	납기 지연	
1	197.3	1	6	18	17
2	13.28	4	23	42	38
3	10	10	24	57	47
4	0	0	49	46	46
5	0	3	43	54	51
6	22	4	30	69	65
7	0.02	0	32	41	41
8	0	0	51	62	62
9	0.02	0	51	21	21
10	10	0	24	23	23
11	0	0	45	26	26
12	113	0	19	33	33
13	13	0	16	54	54
14	89	0	46	7	7
15	66	2	40	25	23
16	155	3	23	38	35
17	110.01	0	26	20	20
18	20.05	2	16	41	39
19	52	0	7	54	54
20	52	0	21	10	10
21	23.59	0	36	20	20
22	19.8	2	38	41	39
23	98.8	3	16	28	25
24	80	0	24	17	17
25	67	8	19	34	26
26	38.5	0	25	45	45
27	57.2	0	22	31	31
28	56.8	6	28	23	17
29	112	0	30	24	24
30	78	0	36	36	36
31	134	5	25	32	27

32	87	1	19	28	27
33	19.4	2	11	8	6
34	110	0	23	19	19
35	156	6	46	30	24
36	58.25	2	10	40	38
37	112	0	41	50	50
38	12.5	0	25	22	22
39	44	3	43	27	24
40	20.6	0	13	18	18
41	10.15	6	36	43	37
42	99	2	19	19	17
43	56	2	29	31	29
44	55.2	2	12	18	16
45	102	4	49	12	8
46	20	6	27	42	36
47	15	3	27	32	29
48	99.3	2	21	13	11
49	59.07	2	36	35	35
50	40	0	21	33	33
평균	56.64	1.92	27.98	31.84	29.92

유전 알고리즘 성능의 효율성을 보이기 위해 50개의 실험 중 하나의 예를 들어 보았다. 실험을 위한 데이터는 아래에 있는 표 8과 같다. 유전 알고리즘으로 구한 해와 ILOG CPLEX로 구한 해를 각각 Gantt chart로 그린 것이 그림 7, 그림 8이다.

표 8 8×3×8 실험 데이터

공정	작업	1			2			3			납기
		1	2	3	1	2	3	1	2	3	
1	대체기계	1	2	3	2	4	5	1	2	3	10
	가공시간	2	3	4	3	2	4	1	4	5	
2	대체기계	1	3	5	1	2	5	3	5	6	12
	가공시간	3	5	2	4	3	6	4	7	11	
3	대체기계	1	2	2	4	5	3	5	6	13	
	가공시간	5	6	4	3	5	13	9	2		
4	대체기계	1	3	4	2	4	6	1	3	6	15
	가공시간	9	7	9	6	5	5	1	3	3	
5	대체기계	4	5	6	6	7	8	1	2	8	7
	가공시간	7	8	2	1	4	4	7	1	5	
6	대체기계	3	4	5	6	7	8	1	2	8	13
	가공시간	1	3	6	7	4	5	3	6	9	
7	대체기계	3	6	4	7	8	1	6	8	6	
	가공시간	2	8	4	1	5	12	7	1		
8	대체기계	2	3	4	3	7	8	4	5	8	13
	가공시간	6	6	7	7	3	2	4	5	9	

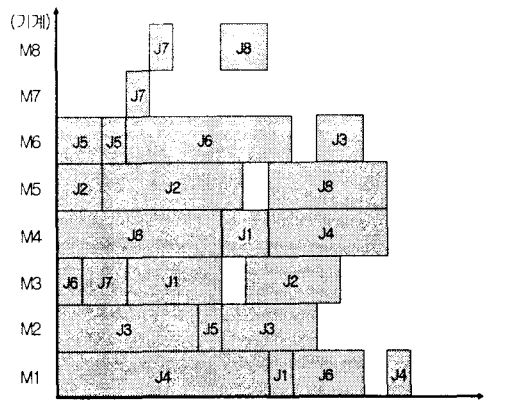


그림 7 8×3×8 문제의 실험결과(ILOG CPLEX)

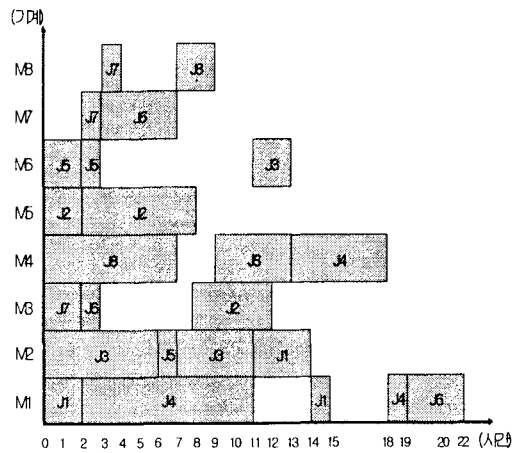


그림 8 8×8×3 문제의 실험결과(GA)

### 5. 결론 및 추후 연구과제

대부분의 중소 제조업체들은 지금 사용하고 있는 job shop 형태의 생산시스템을 그대로 유지하면서 다양한 변화에 적응해가야만 하는 실정이다. 그러나 전통적인 job shop 제조환경에서 제시하는 가정들은 현실적으로 불합리한 경우가 많다. 본 연구에서는 이러한 가정들을 완화시켜 각 공정을 처리할 수 있는 기계가 적어도 한 대 이상 있는, 즉 각 공정에 대체기계가 존재한다고 가정함으로써 다양한 변화에 적용할 수 있도록 하였다.

제조업체에서의 중요한 부분 중의 하나는 최대한 빠르게 고객의 요구인 납기일에 맞춘 생산 스케줄을 산출하는 것이다. 이와 같은 관점에서 볼 때, 공정에 대한 대체기계가 있는 것은 고객의 요구인 납기를 맞추는데 효율적인 방법이 될 수 있

을 것이다.

본 연구는 각 작업의 공정을 수행하는 여러 대의 대체기계가 존재하는 job shop 일정계획 문제에서 유전 알고리즘을 이용한 발견적 기법의 사용이 비교적 빠르게 해를 산출한다는 것을 입증하였다. 이는 시간과 자원의 제약으로 변화에 대처하기 힘든 중소 업체들에게 빠르게 결과를 제공할 수 있을 것이다. 따라서 본 연구에서 제시한 일정계획 방법은 실제 현장에서 효율적인 일정계획을 수립하는데 충분히 활용될 수 있을 것이라고 생각한다.

추후 연구과제로는 첫째, 작업장의 유연성을 늘리려고 대체기계를 도입할 경우에, 실제로 얻는 이득보다 비용이 더 많이 들어갈 가능성도 있을 수 있다. 따라서 대체기계 도입에 관한 타당한 경제성 평가가 뒤따라야 할 것이다. 둘째, 본 연구에서 대체기계만을 다루었지만, 외주작업이나 초과근무 등과 같은 다른 대안들도 함께 고려하여 연구를 해볼 필요가 있을 것이다. 셋째, 본 연구에서는 유전 알고리즘을 적용시켰지만, 다른 많은 발견적 기법들도 대체기계가 있는 일정계획 문제에 적용시켜 볼만한 충분한 가치가 있을 것이다.

### 참고 문헌

- [1] 강용혁, 이홍철, 김성식, "서로 다른 납기를 갖는 작업에 대한 이중 병렬기계에서의 일정 계획수립", *대한산업공학회지*, 제24권, 제1호, pp.37-50, 1998.
- [2] 김기동, *공정계획과 일정계획의 통합에 관한 연구*, 박사학위 논문, 서울대학교 산업공학과, 1997.
- [3] 김상천, 김정자, 이상완, 이성우, "유연 Job Shop 일정계획의 유연성에 대한 시뮬레이션", *한국산업융용학회 논문집*, 제4권, 제3호, pp.281-287, 2001.
- [4] 김정자, 김상천, "Flexible Job Shop 일정계획을 위한 혼합유전알고리즘의 적용", *대한설비관리학회지*, 제4권, 제4호, pp.149-158, 1999.
- [5] 박병주, 최형림, 김현수, "유전알고리즘에 기반한 Job Shop 일정계획 기법", *경영과학*, 제20권, 제1호, pp.51-64, 2003.
- [6] 정대영, 김기동, 장성용, 박진우, "생산능력 조정을 고려한 Job Shop 일정계획 문제에 관한 연구", *대한산업공학회/한국공업경영학회 99 추계공동학술대회*, pp.306-309, 1999.
- [7] Baker, K. R., *Introduction to Sequencing and Scheduling*, John Wiley & Sons Inc., 1974.